REMARKS/ARGUMENTS

In the final Office action dated September 28, 2005 the Examiner rejected claims 1 - 4, 7 - 9, 12 and 14 - 27 under 35 U.S.C. § 103. By this Amendment and the accompanying Request for Continued Examination, Applicant has amended claim 1 and added claims 28 and 29. Reconsideration and reexamination are requested for claims 1 - 4, 7 - 9, 12 and 14 - 29 that are now pending in this application.

Claims 1 - 4 and 9, 12 and 14 - 27 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Hobson et al., U.S. Patent No. 6,209,016 (hereafter referred to as "Hobson") in view of Fischer et al., U.S. Patent No. 6,237,016 (hereafter referred to as "Fischer"). Claims 7 and 8 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Hobson in view of Fischer as applied to claim 1 and further in view of Curiger et al., U.S. Patent No. 6,064,740. Claims 1, 21 and 22 are independent. Claims 2 - 4, 7 - 9, 12, 14 - 20 and 23 - 29 depend on independent claim 1.

Applicant has amended claim 1 in an attempt to further clarify the previously discussed distinctions between the claim and Hobson and Fischer. Applicant respectfully submits Hobson and Fischer considered either independently or in combination do not teach or suggest all of the limitations of claims 1. For example, neither Hobson nor Fischer teach or suggest a decode unit and an execution unit that cooperate as claimed in claim 1. Here, the execution unit is "configured to execute arithmetic instructions to perform product and square operations" and the decode unit is "configured to issue the arithmetic instructions to the execution unit so that the execution unit performs specified multiplication and addition operations in parallel and performs specified multiplication operations in parallel."

Hobson discloses a co-processor that performs modular multiplication using various registers, multipliers and adders. See, Hobson Abstract and Figure 2. Hobson teaches that the co-processor is directly controlled by a CPU that executes instructions. See Hobson at column 1, lines 40 - 45 and at column 8, lines 25 - 30:

> The co-processor is directly driven by the microcontroller's CPU under software control by a program stored either in ROM or in EEPROM. Such a co-processor which implements the Montgomery algorithm for modular reduction without the division process and is known from European Patent Publication EP-0601907-A.
>
> Appropriate sequencing of events to allow the new co-processor to perform this calculation are described below. The control of the sequence may be under software control using the CPU. In this case the CPU overhead is now minimal, otherwise control of the sequence of calculations may be done using a dedicated hardware state machine.

Hobson also teaches that the CPU performs an operation of determining whether to perform a modular square or a modular multiply. See Hobson at column 6, lines 44 - 56:

> In the known co-processor, in order to perform exponentiation operations as required for RSA Public Key systems, the CPU has to regulate the exponentiation process under software control by examining each exponent bit in sequence. The current bit is used to decide whether to perform a modular square or a modular multiply. The exponent value is stored in memory and is read by the CPU one byte at a time as needed. The current bit value is determined by an instruction sequence. As the co-processor requires the CPU to provide the A value during the modular operation, the determination of the exponent bit can only happen between modular operations. Only then can the CPU control the co-processor mode of operation.

Hobson does not teach or suggest that a decode unit as claimed (e.g., "configured to determine if a square operation or a product operation needs to be performed on an operand") is "configured to issue the arithmetic instructions to the execution unit" where the execution unit is "configured to execute arithmetic instructions to perform product and square operations" as claimed. In Hobson instructions are only executed by the CPU. Instructions are not passed to the co-processor that performs the modular multiplication. Rather, the CPU "directly controls" the operation of the co-processor.

This latter point is further evidenced by the inputs to the co-processors shown in Figures 1 and 2. All of the inputs to these co-processors are data inputs (e.g., B, D, A, etc.). See, for example, Hobson at column 5, line 55 - column 6, line 6 and at column 8, lines 43 - 56. Hobson does not teach or suggest that the co-processor could or advantageously should receive instructions.

Fischer discloses a method an apparatus for performing complex digital filters. Fischer, Abstract. In Figure 1, a processor 105 includes a decode unit 104 and an execution unit 142. The operations of and interactions between the decode unit and the execution unit are discussed at column 7, lines 23 - 41 and column 8, lines 36 - 41:

> FIG. 1 additionally illustrates that the processor 105 includes a decode unit 140, a set of registers 141, an execution unit 142, and an internal bus 143 for executing instructions. Of course, the processor 105 contains additional circuitry, which is not necessary to understanding the invention. The decode unit 140, registers 141 and execution unit 142 are coupled together by internal bus 143. The decode unit 140 is used for decoding instructions received by processor 105 into control signals and/or microcode entry points. In response to these control signals and/or microcode entry points, the execution unit 142 performs the appropriate operations. The decode unit 140 may be implemented using any number of different mechanisms (e.g., a look-up table, a hardware implementation, a PLA, etc.). While the decoding of the various instructions is represented herein by a series of if/then statements, it is understood that the execution of an instruction does not require a serial processing of these if/then statements. Rather, any mechanism for logically performing this if/then processing is considered to be within the scope of the implementation of the invention.
>
> FIG. 2B illustrates a circuit for the multiply-add instruction according to one embodiment of the invention. A control unit 240 processes the control signal for the multiply-add instruction. The control unit 240 outputs signals on an enable line 242 to control a packed multiply-adder 244.

Fischer thus teaches that the decode unit executes the instructions. Fischer does not teach or suggest that the decode unit issues instructions to an execution unit that performs the multiply-add operation. Accordingly, Fischer does not teach or suggest a decode unit and an execution unit that cooperate as claimed in claim 1 where the execution unit is "configured to execute arithmetic instructions to perform product and square operations" and the decode unit is "configured to issue the arithmetic instructions to the execution unit so that the execution unit performs specified multiplication and addition operations in parallel and performs specified multiplication operations in parallel."

Claim 21 and 22 also include limitations relating to a decode unit issuing instructions to an execution unit. For example, claim 21 recites, in part: "receiving, by a decode unit, a request to perform a modular operation; determining, by the decode unit, whether a Montgomery square operation or a Montgomery product operation is to be performed; issuing, by the decode unit, a first instruction to perform a Montgomery square operation; . . . performing, by an execution unit, simultaneous multiplication operations in response to at least one of the first instruction and the second instruction." Claim 22 recites, in part "determining, by a decode unit, whether to perform a Montgomery square operation or a Montgomery product operation; issuing, by the decode unit, a first set of instructions for an execution unit to perform the Montgomery square operation."

In view of the above Applicant submits that the cited references do not teach or suggest claim 1, 21 or 22. Claims 2 - 4, 7 - 9, 12, 14 - 20 and 23 - 29 that depend on claim 1 also are patentable over the cited references for the reasons set forth above. In addition, these dependent claims are patentable over these references for the additional limitations that the dependent claims contain.

For example, the cited references do not teach or suggest performing multiplication and addition operations in parallel in a clock cycle as set forth, for example, in claims 23 and 24. The cited references teach performing the operations in multiple cycles. For example, Fischer teaches at column 6, lines 13 - 15 that a multiply-accumulate operation requires two instructions. Hobson teaches that data is clocked and manipulated one or two bits at a time. See, for example, column 2, lines 28 - 48 (registers are "serially clocked one bit at a time" for multiplication operations and summing operations are performed "over the next 544 clock cycles") and column 4, lines 23 - 27 (the improvement involves processing two bits at a time per clock period) and column 5, lines 23 - 35 (the product is generated over 512 clock cycles).

With regard to claim 29, the cited reference do not teach or suggest issuing "a plurality of types of add-subtract instructions and a plurality of types of multiply instructions" to an execution unit as claimed.

CONCLUSION

In view of the above remarks, Applicant submits that the claims are patentably distinct over the cited references and that all the rejections to the claims have been overcome. Reconsideration and reexamination of the above Application is requested.

Respectfully submitted,

CHRISTIE, PARKER & HALE, LLP

By _____

Stephen D. Burbach
Reg. No. 40,285
626/795-9900

SDB/sdb

SDB PAS664629.1-*-01/30/06 10:50 PM